Rule WLM032: Server was assigned CPU protection but most work done in support of lower importance work

Finding: A server (CICS or IMS region) was assigned long-Term CPU Protection, but most CPU work in the CICS or IMS region was done in support of lower importance transaction service classes.

Impact: This finding should be viewed as MEDIUM IMPACT or HIGH IMPACT on the performance of your computer system. The level of impact depend on the amount of CPU activity that is given to lower importance work, the importance of the work that might be denied CPU service because CPU protection was given to work processed by the server, and whether management is concerned about mis-allocation of CPU resources.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: With OS/390 Release 10 (and with APAR OW43855 installed), IBM introduced the Long-Term CPU Protection option for Goal Mode. When Long-Term CPU Protection is assigned to a service class, the Workload Manager attempts to ensure that less important service class periods will have a lower dispatch priority than the service class that is assigned Long-Term CPU Protection.

The Long-Term CPU Protection option was implemented in OS/390 Release 10 - long after Goal Mode was announced. This option was released so late after Goal Mode was announced because CPU Protection significantly deteriorates the basic concepts of Goal Mode design.

One reason that IBM introduced Goal Mode as a way of minimizing the requirement that systems personnel have detailed knowledge of system-internals. Prior to Goal Mode, system programmers and performance analysts needed to understand the detailed internal logic of the System Resources Manager (SRM), so they could make specific changes to the IEAIPSxx and IEAOPTxx members of SYS1.PARMLIB to obtain good performance of their computer systems. Sadly, individuals with such detailed knowledge were scarce and organizations were not always able to obtain good performance because of the lack of knowledge.

With IBM's plans for increasingly complex environments (parallel sysplex, coupling facility, etc.), IBM realized that an increased level of knowledge would be required to optimize performance for these environments. IBM designed the Workload Manager (WLM) as a way to automate the

performance tuning process, and reduce the level of knowledge required to manage systems performance.

The basic concept of the Workload Manager is that installation personnel should:

- Stratify their work into several levels of importance,
- Set performance goals for work at each level of importance, and
- Let the system automatically allocate resources so the performance goals would be met and the optimum amount of work would be performed.

IBM SRM/WLM developers designed complex algorithms to ensure that goals were met for important work, and that system resources were distributed to all work to optimize throughput. In most instances, the algorithms work extremely well, if the goals are set to match management or user expectations, and the importance of work is correctly specified.

There are, however, a few design decisions that can cause performance problems for important work, or can cause management consternation.

- **Small CPU consumer**. With Goal Mode, the SRM/WLM developers introduced the "small CPU consumer" algorithm. The intent of the small CPU consumer algorithm was to identify those few tasks that (1) used a very small amount of CPU and (2) would quickly finish if given a modest amount of CPU time. The WLM attempted to identify these small CPU consumer tasks, given them a high priority access to a CPU, and expect that the tasks would quickly finish and be removed from the system. When Goal Mode was first introduced, the WLM gave these small CPU consumer tasks very high dispatching priority (the dispatching priority was 254; right below Master (with 255), and above all other work).

  Giving small CPU consumers such high CPU dispatching priority was not popular, since the CPU dispatching priority of unimportant batch work could be higher than the CPU dispatching priority of SYSSTC (system started tasks). Consequently, IBM changed the CPU dispatching priority scheme to be 255 (Master), 254 (SYSSTC), and 253 (small CPU consumers).

  The resulting scheme still was not popular, because this scheme meant that the CPU dispatching priority of small CPU consumer work would be higher than work with goals (work with goals is the production work that has performance goals and importance assigned). Sadly, the small CPU consumers often consisted of low-importance work (such as short batch jobs), and installation personnel would see unimportant work have a

higher CPU dispatching priority than the most important production work. Complicating the issue was that the WLM would not always recognized in a timely fashion that work that had been identified as “small CPU consumers” had, in fact, changed their processing characteristics and were no longer small CPU consumers. Thus, the “small CPU consumers” would have a negative effect on the performance of critical work with goals. For a very few users, this negative effect (even though it lasted for a very short time) would so severely effect their operation that they would go back to Compatibility Mode.

- **Allocation of CPU dispatching priority**. The WLM attempts to maximize throughput, while meeting performance goals. As a part of the resource allocation algorithms, the WLM can determine that a particular service class period is missing its goal because it is being denied access to a CPU. In this case, the WLM can examine the CPU use of service class periods with a higher dispatching priority to see whether those service class periods would be harmed if dispatching priorities were realigned. The WLM can decide that there would be no detrimental effect on the service class periods with a higher CPU dispatching priority, if the service class missing its goal were placed above the service class periods with a higher CPU dispatching priority. These algorithms are very conservative, in that they analyze history to make sure that the higher importance work was always exceeding its goal, that the higher importance work would not be denied CPU access and fail to meet its goal if the lower importance work had a higher CPU dispatching priority, and that assigning a higher CPU priority to the lower importance work would help the lower importance work meet its goal.

  The WLM concept is that a lower importance service class needs access to a CPU and that giving this access would not cause a higher importance service class service class to miss its goal. In most cases, this concept works well.

  There have been, however, situations in which the algorithms did not work well because one of the service class periods (either the low importance work or the high importance work) changed its processing characteristics. If the low importance work with a higher CPU dispatching priority suddenly began requiring more CPU, the WLM would take action to reduce its CPU dispatching priority. Unfortunately, the WLM requires some elapsed time to notice that the situation has changed and to take remedial action. During this elapsed time, performance could suffer for the high importance work.

- **Management consternation**. Regardless of whether work with a high importance met its goal, management often becomes concerned if they observe lower importance work having a CPU dispatching priority higher

than high importance work. Management often feels that there is something intuitively wrong with the situation, and do not approve.

For these reasons, some very important IBM users adamantly demanded that the WLM CPU dispatching priority allocation algorithms be changed. Consequently, IBM introduced the Long-Term CPU Protection option. As mentioned earlier, when Long-Term CPU Protection is assigned to a service class, the Workload Manager attempts to ensure that less important service class periods will have a lower dispatch priority than the service class that is assigned Long-Term CPU Protection.

In general, implementing Long-Term CPU Protection is not a good idea. This is because (except in very special situations) Long-Term CPU Protection automatically removes much of the flexibility in the WLM algorithms as they attempt to allocate resources so that all service class periods meet their performance goals, and as they attempt to maximize throughput. **Only if you have very time-critical work that cannot tolerate any occasional delay for CPU, should you implement Long-Term CPU Protection!**

That said, there is one additional aspect of Long-Term CPU Protection that must be understood: if Long-Term CPU Protection is implemented for a server (CICS or IMS region), the CPU protection applies to **all work** processed by the server, regardless of the importance of the associated transaction service classes.

It is common for a server to process transactions that have been assigned to more than one transaction service class, since some transactions are very important while other transactions are less important. The transactions can be classified and assigned to different transaction service classes, and the transaction service classes can have different performance goals and goal importance. If transactions in these different transaction service classes are processed by a server with CPU protection assigned, all transactions receive the same CPU protection, regardless of their importance.

CPExpert examines variable R723MSCF (Service/Report class flags) in SMF Type 72 information to detect whether Long-Term CPU Protection has been implemented for a service class. CPExpert further determines whether that particular service class is a server (CICS or IMS region) for more than one transaction service class and whether the transaction service classes have different goal importance. If this is the case, CPExpert sums the *service* given to transaction service classes.

The *service* given to a transaction service class by a server is extracted from the “Service Class Served Data Section” of SMF Type 72 records. The total service provided by the server to all transaction service classes

is summed, and a percent of service given to each transaction service class is calculated.

CPExpert separately sums the service given to transaction service classes between the service provided to the transaction service classes at the highest goal importance, and the service provided to all other transaction classes at a lower goal importance.

CPExpert produces Rule WLM032 when the total service provided to the transaction service classes at the highest goal importance is less than the value specified for the **PCTSERVC** guidance variable in USOURCE(WLMGUIDE). The default value for the PCTSERVC guidance variable is 50%, indicating that Rule WLM032 would be produced when the most important transaction service classes received less than 50% of the service provided by the server to all transaction service classes.

The following example illustrates the output from Rule WLM032:

```
RULE WLM032: SERVER PROTECTED BUT MOST WORK DONE FOR LOWER IMPORTANCE

   Service Policy WLMPOL01 (effective 12SEP2002:14:20:20) specified CPU
   Protection for the CICSRGN service class.  The CICSRGN service class
   was a server for transaction service classes that had different Goal
   Importance levels.  Some of the work done by the CICSRGN service class
   was in support of a high Goal Importance, but more than 50% of the work
   supported transaction service classes with a low Goal Importance.  The
   below information shows how often CICSRGN provided service to transaction
   service classes at different importance:

                              TRANSACTION        GOAL          PERCENT
   MEASUREMENT INTERVAL      SERVICE CLASS    IMPORTANCE       SERVICE
   14:45-15:00,03OCT2002       CICS               2              1.7
   14:45-15:00,03OCT2002       CICSDEFA           3             49.0
   14:45-15:00,03OCT2002       CICSLONG           3             47.3
   14:45-15:00,03OCT2002       CICSCONV           3              2.1
   14:45-15:00,03OCT2002       CICSMISC           3              0.0
```

Suggestion: Rule WLM032 alerts you to the potential resource allocation problem resulting from low importance work receiving most of the service provided by the server (CICS or IMS region). Since Long-Term CPU Protection was specified for the server service class, this low importance work is processed at a high CPU dispatching priority. Since the work is low importance, you should consider whether management wishes this low importance work to receive the favorable CPU protection. If Rule WLM032 is produced regularly, CPExpert suggests that you consider the following alternatives:

- Remove Long-Term CPU Protection from the CICS or IMS server service class. In most cases, this is the best alternative. As explained above, the Long-Term CPU Protection option is suited only for very special

cases. For the normal case, the WLM resource allocation algorithms will provide adequate CPU dispatching priority for work with a high importance. If that work should miss its performance goal, the WLM will adjust resources (including CPU dispatching priority) as necessary to ensure that the work achieves its performance goal.

- Revise the work classification scheme to remove the low importance work from the CICS or IMS server, and assign these transactions to a CICS or IMS server that does not have the Long-Term CPU Protection option enabled. This alternative might require that a separate CICS or IMS region be established, or might be possible using existing regions that serve other low importance transactions.

- You can alter CPExpert's analysis by modifying the PCTSERVC guidance variable in USOURCE(WLMGUIDE).

- You can "turn off" Rule WLM032 by specifying **%LET WLM032 = OFF.**; in USOURCE(WLMGUIDE) if decide not to implement I/O Priority Management.

- You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is specified as **%LET CHCKPLCY=Y;**, CPExpert will not check the service definition for potential problems.

  Before you globally disable CPExpert's checking the service definition, you may wish to review other guidance variables. Many of the tests which CPExpert makes can be made inoperative by a guidance variable that applies to the specific test. The discussion of each finding describes the associated guidance variable.

Reference: MVS Planning: Workload Management

OS/390 (V2R10):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R1):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R2):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R3):	Chapter 12.2: Long-Term CPU Protection
z/OS (V1R4):	Chapter 12.2: Long-Term CPU Protection

